REMARKS

The following remarks are deemed fully responsive to the outstanding final office action mailed June 27, 2005. Claims 1, 9, 12, 13 and 15 are amended for clarity and without new matter; for example, limitations of claim 11 are added to claims 1 and 9; limitations of claim 14 are added to claims 12 and 13; limitations of claim 17 are added to claim 15. Claims 11, 14 and 17 are thus cancelled. Claim 18 is newly added; support for claim 18 is found at least in paragraphs 5, 14 and 17 and in FIG. 4 of the diagrams and in claims 1, 15. Claims 1 – 10, 12, 13, 15, 16 and 18 are now pending, of which claims 1, 12, 15 and 18 are independent.

Claim Rejections – 35 U.S.C. § 102

Claims 1-14 stand rejected under 35 U.S.C. § 102(e) as being taught by U.S. Patent Number 6,269,437 granted to Batten et al., (hereinafter "Batten") *(note, as in the previous office action, the final office action states 6,269,439, which we believe is a typographical error)*. Applicant respectfully disagrees.

To anticipate a claim, Batten must teach every element of the claim and "the identical invention must be shown in as complete detail as contained in the ... claim." *MPEP 2131* citing *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, 2 USPQ2d 1051 (Fed. Cir. 1987) and *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913 (Fed. Cir. 1989). Batten does not teach every element of claims 1-14.

The immediate application "processes bundles of instructions preferentially through clusters such that bypassing is substantially maintained within a single cluster." See paragraph [0004] of the specification. Paragraph [0014] of the specification, for example, recites "decode unit 130 detects and then distributes bundled instructions to the program counters 104 according to the threads associated with the instructions." The specification continues that in a wide mode "bundled instructions from the same thread are processed through multiple clusters 102 at the same time," and that in a throughput mode "bundled instructions from one thread are processed through one program counter 104, and through a corresponding cluster 102; bundled instructions from other threads are likewise processed through another program counter and cluster pair 104, 102." Thus, the processor of the immediate application clearly has two modes of operation: (1) throughput mode, processing one

thread per cluster, and (2) wide mode, processing one thread through multiple clusters. Bypassing between clusters can be reduced by processing bundled singly-threaded instructions within a cluster, thereby avoiding delays associated with bypassing between clusters.

On the other hand, Batten is concerned with "reducing port pressure of clustered microprocessors." See Batten col. 4, lines 65-66. Batten does not disclose, or even suggest, two modes of operation, the bundling of instructions, or the processing of instructions from one thread through one cluster. In fact, Batten does not disclose or even suggest threads. Further, Batten does not disclose any specific method of processing instructions at all, let alone bundling instructions *in association with threads*.

In the Examiner's argument of paragraph 28, page 9 of the final office action, in essence the Examiner asserts that since Batten's device fetches instructions from a program in memory (Batten column 11, lines 1-15 and Figure 12), the instructions are singularly threaded. Respectfully we disagree. First, Batten does not use the word 'program' anywhere. Secondly, Dictionary.com defines *thread* as: *A portion of a program that can run independently of and concurrently with other portions of the program.* As known in the art, a program or application may have multiple *threads* executing in a shared address space. Threads are thus sub-processes that share code and data segments but have their own program counters, registers and stack. The Examiner's assumption that Batten's device is fetching singly-threaded instructions is therefore not substantiated.

The immediate application and claims specifically addresses processing of bundled singly-threaded instructions. Batten has no teaching of threads or bundling of singly-threaded instructions and cannot, therefore, anticipate the claims.

In particular, amended claim 1 recites a method for processing bundled instructions through execution units of a processor, including the steps of:

a) determining a throughput mode of operation, based upon a configuration bit;
b) fetching a first bundle of singly-threaded instructions from a singly- or multiply-threaded program;
c) distributing the first bundle to a first cluster of the execution units for execution therethrough;

d) fetching a second bundle of singly-threaded instructions from the program; and

e) distributing the second bundle to a second cluster of the execution units for execution therethrough.

As noted above, Batten has no teaching of threads or bundling of singly-threaded instructions. Step a) of claim 1 requires that a throughput mode of operation is determined from a configuration bit. Batten does not disclose the use of a configuration bit for determining a mode of operation. In FIG. 12, Batten shows a memory 102 connected to a fetch unit 104; however Batten does not disclose reading of instructions from the memory, nor bundling singly-threaded instructions as required by steps b) and d).

We must therefore assert, respectfully, that the Examiner mistakenly uses hindsight to interpret operation of Batten's memory and fetch unit by suggesting that singly-threaded instructions are bundled. But, Batten does not disclose or suggest distributing bundled instructions to a specific cluster, as required by steps c) and e). Further, Batten does not disclose fetching a second bundle of singly-threaded instructions, as required by step d). Batten further does not disclose executing singly-threaded instructions on a specific cluster, as required by step e).

Notably, Batten makes no distinction between processing of threads and does not teach, suggest or disclose bundling of singly-threaded instructions. Batten, therefore, cannot anticipate claim 1. Reconsideration of claim 1 is respectfully requested.

Claims 2-11 depend from claim 1 and benefit from like argument. However, these claims have additional features that patentably distinguish over Batten. For example, claim 2 recites processing the first bundle within the first cluster. Claim 3 recites processing the second bundle within the second cluster. As argued above, Batten does not disclose processing bundled instructions within a cluster.

Claim 7 recites decoding instructions into the first bundle of the singly-threaded instructions. Claim 8 recites decoding instructions into the second bundle of the singly-threaded instructions. Batten does not disclose decoding singly-threaded instruction into a first or second bundle, and therefore cannot teach limitations of claims 7 and 8.

Amended claim 9 recites selecting the configuration bit to specify a wide mode of operation, fetching a third bundle of singly-threaded instructions from the program, distributing the third bundle to the first and second clusters of the execution units for execution therethrough and bypassing data between the clusters, as needed, to facilitate processing of the third bundle through the clusters. Again, Batten does not disclose the use of a configuration bit for determining a mode of operation and does not teach fetching, distributing or bypassing data between clusters for bundled instructions. Therefore, Batten cannot anticipate claim 9.

Claim 11 recites selecting a configuration bit prior to the steps of fetching the third bundle, distributing the third bundle, and bypassing data between the clusters. Batten does not disclose a configuration bit for determining a mode of operation, and, as argued above, does not disclose bundles of instructions.

In view of the above arguments, Batten does not anticipate any of claims 2-11. Reconsideration of claims 2-11 is respectfully requested.

Amended claim 12 recites a method for processing bundled instructions through execution units of a processor, including the steps of:

a) determining a wide mode of operation, based upon a configuration bit;
b) fetching a first bundle of singly-threaded instructions from a singly- or multiply-threaded program;
c) distributing the first bundle to two or more clusters of the execution units for execution therethrough; and
d) bypassing data between the clusters, as needed, to facilitate processing of the first bundle through the clusters.

As argued above, Batten does not disclose determining a mode of operation based upon a configuration bit, as required by step a) of claim 12. Batten does not disclose or suggest fetching a first bundle of singly-threaded instructions, as required by step b). Further, Batten does not disclose or suggest distributing bundled instructions to two or more clusters, as required by step c). Batten, therefore, cannot anticipate claim 12 for at least these reasons. Reconsideration of claim 12 is respectfully requested.

Claim 13 depends from claim 12 and benefit from like argument. However, these claims have additional features that patentably distinguish over Batten. For example, claim 13 recites selecting the configuration bit to indicate a throughput

mode of operation, fetching a second bundle of singly-threaded instructions from the program, distributing the second bundle to one of the clusters for execution therethrough, fetching a third bundle of singly-threaded instructions from the program, and distributing the third bundle to another one of the clusters units for execution therethrough. As argued above, Batten does not disclose or suggest fetching a bundle of singly-threaded instructions or of distributing bundles of instructions to clusters. Therefore, Batten cannot anticipate claim 13. Reconsideration of claim 13 is respectfully requested.

Claims 15 and 16 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Number 5,729,761 granted to Murata et al. (hereinafter "Murata"). Applicant respectfully disagrees. Murata does not teach every element of claims 15-17.

Murata discloses a system for upgrading a program executed by one cluster of a plurality of clusters. Murata discloses that the plurality of clusters operate in a restricted mode, whereby one cluster may be upgraded while allowing the remaining clusters to continue operation. Murata does not however disclose a selectable mode of operation for the clusters whereby throughput is improved or a thread's performance is improved.

In particular, amended claim 15 recites a processor architecture of the type having two or more clusters of execution units for processing instructions, the improvement comprising:

a) a configuration bit for specifying a wide mode or a throughput mode of operation;

b) a thread decoder for grouping instructions of a singly- or multiply-threaded program into singly-threaded bundles and for distributing the bundles to the clusters according to the configuration bit;

c) wherein the singly-threaded bundles are distributed across a plurality of clusters in the wide mode and each singly-threaded bundle is distributed to one of the clusters in throughput mode.

Murata does not disclose or suggest a configuration bit for specifying a wide mode or a throughput mode of operation as required by step a) of claim 15. Murata also does not disclose or suggest a thread decoder for grouping instructions into singly-threaded bundles and for distributing the bundles to the clusters according to

either a wide mode or throughput mode of operation, as required by step b) of claim 15. Murata instead discloses "a cluster non-restricted mode in which the distributed processing of any one job among processors belonging to any cluster is permitted, and a cluster restricted mode in which the distributed processing of one job among processors belonging to different clusters is restricted." See Murata col. 1, lines 57-61. Moreover, Murata does not teach use of singly-threaded bundles of instructions.

In short, Murata cannot anticipate claim 15. Reconsideration of claim 15 is respectfully requested.

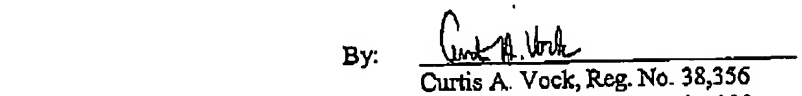
Claim 16 depends from claim 15 and benefit from like argument. Reconsideration is requested for claim 16.

New claim 18 contains features of claims 1, 15 which, again, are not taught or suggested by either Batten or Murata; allowance of claim 18 is therefore respectfully requested.

Applicant authorizes the Commissioner to charge a fee of $200 to Deposit Account No. 08-2025 for additional independent claim 18.

Applicant believes no other fees are due in connection with this Response; however, if any additional fee is deemed necessary, the Commissioner is authorized to charge such fee to Deposit Account No. 08-2025.

Respectfully submitted,
LATHROP & GAGE, LC.

By: 

Curtis A. Vock, Reg. No. 38,356
4845 Pearl East Circle, Suite 300
Boulder, CO 80301
Telephone: (720) 931-3011
Facsimile: (720) 931-3001